DETECTING TEXTUAL SALIENCY IN PRIVACY POLICY

DS5500 PROJECT REPORT - PHASE 1
NORTHEASTERN UNIVERSITY

Kaushik Nishtala
nishtala.k@northeastern.edu

Shubhanshu Gupta
gupta.shubh@northeastern.edu

November 10, 2021

ABSTRACT

Websites, mobile apps, and other product and service providers share how they gather, use and manage customers' data in the form of privacy policy documents. However, due to their lengthy and complex nature, a majority of people tend to ignore these documents. To counteract the risks posed by this ignorance, we present preliminary language models to extract and deliver salient policy information to end-users in the first phase of the project. The findings show that our baseline models provide promising results and set the direction for the second phase by motivating the need for adding model and data complexities. In addition, we discuss the implementation of the interactive web framework as an extension to the results produced from the language models.

1 Introduction

Data privacy is a matter of importance and concern for an average consumer. It is the right of any individual to have control over how their personal information is collected and used. However, with the advent of technology and social media, any trivial activity (browsing a webpage, using a product or service, or even a mobile app) allows companies to directly or indirectly track users' data. The collected data is then used for various purposes such as advertising, marketing, or research, to name a few. In order to regulate this process, government and regulatory bodies adopt privacy regulations that force these companies to share how they gather, use, and manage users' data. This regulation is primarily accomplished in the form of a privacy policy document.

However, the abundance of information in these documents makes them quite lengthy and challenging for lay users to comprehend their contents. In fact, some findings [9] state that more than 85% of privacy policies scraped are at or above college-level reading difficulty. As the governmental bodies increasingly adopt regulations to promote transparency, among other things, it brings an opposite effect in driving users to ignore the policies altogether due to their wordiness and intricacy. To counter this ignorance of these documents and make it easier for users to understand them better, we use machine learning algorithms and natural language processing to derive salient information from these documents to end-users. We do this in the form of the INFORM and the QUERY modules.

2 Objective

The primary objective of the INFORM module lies in communicating a high level and a more granular breakdown of privacy policies. The functionality is twofold. The first one is an interactive visualization framework powered by Tableau, Flask, and SQLAlchemy which enables users to explore the trends and patterns in privacy policy practices interactively. The other part of the module involves building a selection of models that can classify privacy policy paragraphs into a set of pre-defined privacy practice categories using supervised machine learning algorithms.

Category	Fleiss' Kappa
First Party Collection/Use	.76
Third Party Sharing/Collection	.76
Other	.49
User Choice/Control	.61
Data Security	.67
International and Specific Audiences	.87
User Access, Edit and Deletion	.74
Policy Change	.73
Data Retention	.55
Do Not Track	.91

Table 1: List of data practice categories in the corpus along with their Fleiss' Kappa values [15].

On the other hand, the QUERY module can enable users to ask policy-related questions using a freeform question-answering system for privacy policies. By retrieving the most relevant information from the policy document given a question, this system facilitates searching the target information from a lengthy policy document. To this purpose, we make use of a Question-Answer dataset that provides 714 human-annotated questions written for a wide range of privacy practices. We finally augment the functionality of both the modules with an interactive web framework that enables users to understand their data privacy rights better.

Organization. The report discusses the methods and functionality pertaining to the INFORM module. The rest of the paper is divided as follows: in section 3, we briefly review existing studies on privacy policies. Section 4 provides technical description of the methods used in the project. In sections 5, a set of preliminary experiments and results is presented and discussed. Finally, section 6 concludes the presented work and suggests future directions to pursue for the following phase.

3 Related Work

Many studies on privacy policy analysis have emerged in light of the General Data Protection Regulation (GDPR). One study that is particularly of interest to us is the Polisis work [4]. According to the paper, the union-based gold standard is used for experiments with Convolutional Neural Networks seeded with domain-specific word embeddings. They scraped, curated, and processed texts of around 130K privacy policies to train these embeddings.

Despite the inspiring work, we believe the study lacks two main elements: They report only the macro-averages and further compute the average of F1 scores on the test set without using a validation set for training. It is also worth noting that a significant amount of information about the training methodology was not shared in the paper, much less about the unbiased performance metrics like the micro averages. As a result, we plan to use their best model performance as a baseline to build and compare our methods. We intend to provide a detailed ablation study of our methods and build models in an effort to replicate or even outperform Polisis with the help of more sophisticated and recently published models and corpus.

4 Methodolody

We present a series of processes, decisions, and artifacts involved in the INFORM module. The process begins with procuring the OPP-115 corpus [15] and performing a slew of pre-processing tasks such as splitting, cleaning, encoding, and tokenizing the policy texts. This is followed by enforcing a data preparation strategy to transform the textual data into a compatible and reliable input format for the machine learning algorithms. Figure 1 shows a flowchart of our methods, each of which is explained in detail in the subsequent sections.

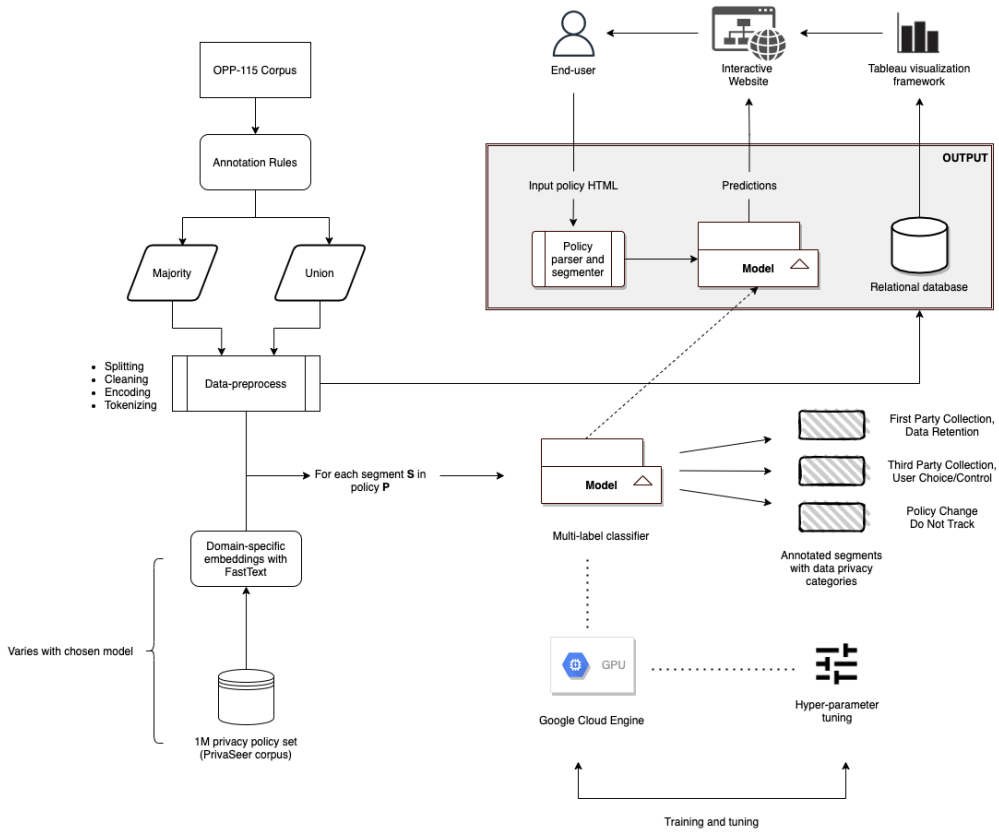

Figure 1: Architecture of the INFORM module: a visual representation of the methods and artifacts involved in the entire framework.

4.1 Data Collection

The curation of the OPP-115 corpus [15] of privacy policies and its annotation scheme produced by domain experts is central to our efforts. An individual data practice in the corpus belongs to at least one of the ten data practice categories, further classified into a set of practice attributes. A data practice, in this context, could be any text span that is associated with one or more of the data practice categories and attributes. Table-1 shows the list of data practice categories along with their Fleiss' Kappa values that indicate the level of agreement between the annotators.

A segment (paragraph) can contain information about multiple categories, such as the First Party Collection/Use, Third-Party Sharing/Collection, etc. According to the corpus curators, the best agreement between the annotators was achieved on the Do Not Track class with Fleiss' Kappa value equal to 91%. On the contrary, the class with the highest disagreement was the Other category, with only 49% of agreement, as shown in Table-1. The Other category was therefore broken down into its attributes: Introductory/Generic, Privacy Contact Information, and Practice Not Covered, forming 12 categories/labels in total.

4.2 Annotation Scheme and Process

A total of 10 skilled annotators applied the annotation scheme to 115 privacy policies as defined by the dataset curators. Accompanied by a web-based tool, the annotators processed each policy divided into paragraph-length segments, one at a time, producing upwards of 23K data practices altogether. Also, each privacy policy was processed and annotated independently by three annotators.

4.3 Data Pre-processing

In obtaining a reliable baseline for privacy policy classification, we compiled two gold standard datasets: union-based, which includes all expert annotations, and the majority-vote-based dataset, where only annotations labeled with a given category by at least two annotators were retained. As shown in Figure 7 (section B.1), in the union-based dataset, all the annotations are taken into account regardless of their frequency. In the case of majority-voting, if at least two annotators labeled any part of a segment with a given category, we label the segment with that particular category.

Additionally, the corpus contains policy information about annotations, segments, and policy metadata spread across several CSV and HTML files. Therefore, as shown in Figure 8 (section B.2), we performed appropriate data pre-processing tasks such as cleaning and merging company-specific annotations, linking related fields, parsing JSON strings, and cleaning dirty site metadata fields.

4.4 Exploration

We established relations between the processed files to create a relational database for powering up the visualization framework. Creating a database allows the framework to link several aspects of the data, allowing users to explore unique patterns and trends in privacy policy practices. The respective Entity Relationship Diagram is shown in Figure 9 (section B.3).

The diagram primarily served as a blueprint for database creation. It comprises seven different tables, each named after its contents. Briefly, the table 'category_metadata' lists different privacy practice categories while the 'attribute_mapping' table shows the many-to-many mappings between the categories and their respective attributes. This mappings table is complemented by another table named 'attribute_metadata' listing the attribute names. Also included is the 'attribute data' table that specifies the attribute values of all the observations (annotations). Furthermore, the table 'site_metadata' maintains company-specific details such as the company name, policy ID, to name a few. A 'site_sector_metadata' table accompanies this for maintaining company and sector mappings since each company can belong to several sectors (Entertainment, Health, etc.). Our master table, 'annotation_data,' has details about all the annotations of privacy policies, annotators, segment text, and foreign keys linking the rest of the tables.

Subsequently, we performed preliminary data analysis to get a sense of the dataset, verify that our data is sufficient for the task at hand, and use it to answer important questions.

4.4.1 How imbalanced is the dataset?

It is essential to know about categories' distribution and what categories make the cut (for performance). Figure 2 shows that we have the categories such as the First Party Collection/Use and the ThirdParty Sharing/Collection in majority while Do Not Track and Data Retention categories are under-represented. This trend indicates that the data imbalance for under-represented categories needs to be handled appropriately.

4.4.2 How many categories are segments, on average, classified into?

Figure 3 shows that we have most segments (around 1852) labeled with just one category, whereas very few are labeled with five or more categories. This is important to consider when splitting the labels into train, validation, and test sets while maintaining the same label proportions across the splits. We discuss this in more detail in the following section.

Visualization framework: We extend the functionality of visualizing the dataset with a web framework that integrates with Tableau, HTML, CSS, and JavaScript on the front-end and is served by Flask and SQLAlchemy on the back-end. The primary purpose of the framework is to convey high-level insights from the corpus to general consumers and regulators by allowing them to filter on metadata fields and attribute values to generate plots for each privacy category. The end-user can browse visualizations that inform trends about data collection practices, data security and retention, user choices and controls, to name a few. An extended example set of questions that we expect the framework to help answer is shown in Appendix A.1.

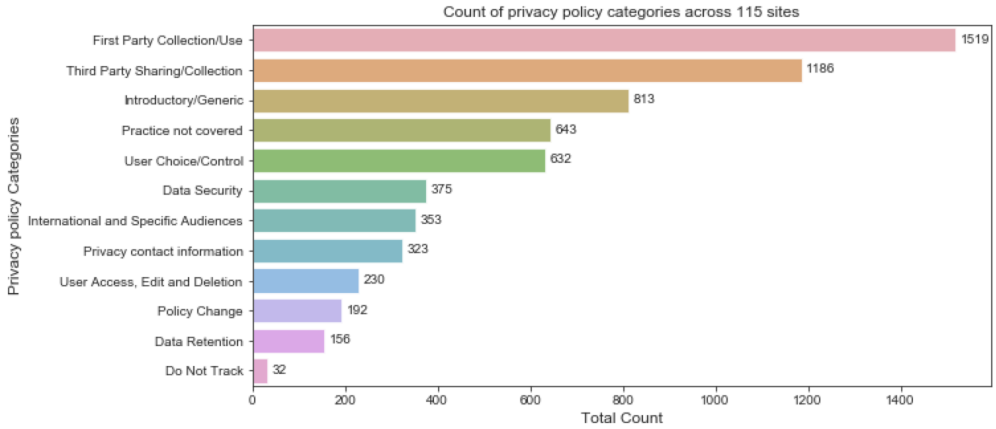

Figure 2: Frequency distribution of privacy policy categories across 115 policies.

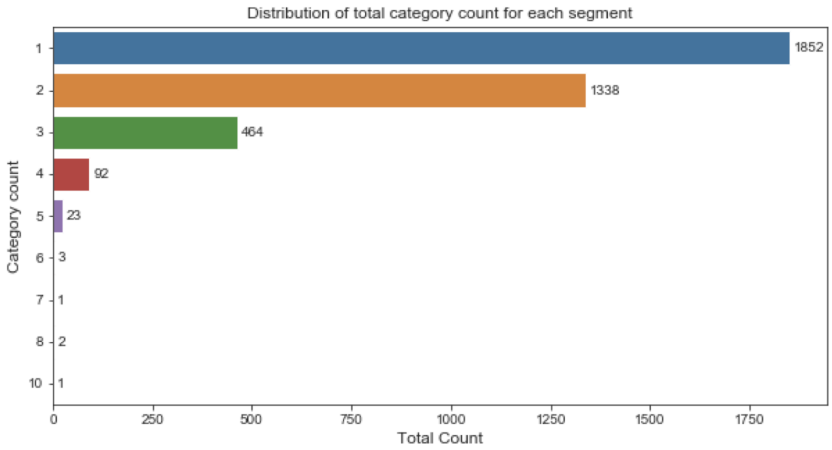

Figure 3: Distribution of total categories labelled for each segment

4.5 Splitting

In order to pursue an unbiased measuring approach, we want to ensure that each data split has similar class distributions after splitting the dataset into train, test, and validation. However, since each observation (segment) can have multiple categories, it complicates the stratification process.

First, we tried splitting the dataset using the traditional way, and as shown in Figure 10 (section B.4), there are significant deviations between the (adjusted) class distributions with a calculated standard deviation of 25.43 across the splits. Note that the train and test proportions are adjusted by balancing the train and test ratios to make it easier to compare them.

To overcome this, we made use of iterative stratification [12] via the skmultilearn library [14], which essentially creates splits while "trying to maintain balanced representation with respect to order-th label combinations". We chose to use an order of one, which provides representative distribution of each category across the splits. Figure 11 (section B.5) shows the adjusted counts for the stratified data splits produced using iterative stratification with a much better standard deviation of 1.83.

4.6 Modeling

4.6.1 Objective

Since each segment can contain information for multiple categories or labels, this problem can be modeled as a multi-label classification problem. An individual classifier is trained on the data to produce probability $P(c_i|x)$, which signifies the occurrence of each category $c_i \in C$, the set of all 12 categories. With the probability of each category as the model prediction, we can determine whether a category belongs to a particular segment by comparing the probabilities to an optimal threshold value.

To choose the model architectures to work with, we considered and prioritized some critical tradeoffs. In the case of performance, we believe that both coarse-grained (overall) and fine-grained (per-category) metrics must be considered to effectively evaluate the model's generalizability. In our case, we think that a model's ability to predict the correct categories accurately (by reducing false positives or precision) is as important as not missing out on any correct categories (by reducing false negatives or recall). Therefore, we consider the primary metric of a model's evaluation to be its F1 score.

However, it is unclear which average scores (micro, macro, or weighted) best define a model's performance. This has been a point of contention in literature [11] where some studies believe that micro average is fair in case of a class imbalance (due to equal weight given to all classes). In contrast, others prefer to use micro-average when there is a class imbalance as it aggregates every class contribution to produce the average. To make fair and unbiased decisions, we report both the averages along with a weighted variant of macro average.

In addition, we would like our models to have a good balance of latency, size, and compute requirements to incorporate into the web framework efficiently. There is no primary motivation for interpretability, but we trust that it could prove helpful for the iterative development of the classifiers. To that end, we report practical model interpretations when possible.

4.6.2 Approach

In this phase of the project, our primary focus is to establish baseline models while motivating the need for adding complexity from both the dataset (e.g., embeddings) and model architecture (e.g., CNN) standpoints. The baselines are beneficial because they enable us to perform ablation study or rapid experimentation via hyperparameter tuning and help discover data or code issues.

One other key motivator to pursue traditional machine learning models is leveraging "the strength of weak learners". The main idea is to build, experiment, and identify models that excel at predicting some aspect of the data irrespective of their overall model performance. We then design an ensemble of these individual models with complementary expertise to create a "mixture of experts". This is a case where "individual classifiers are trained to become experts in some portion of the feature space". This modeling strategy allows us to produce more stable model predictions with each of the individual weak learners bringing unique expertise in predicting some aspect of data.

Traditional ways of representing textual input to a classifier include vector representations like a bag of words or term-frequency inverse-document-frequency (TF-IDF) methods. However, these methods have a clear disadvantage: they do not capture the semantic relations between the words, ordering, and contextual meaning of individual words. Despite their shortcomings, we performed experiments using TF-IDF representations to compare and contrast them with experiments that include higher model and data complexities.

4.6.3 Baselines

Traditional models We built binary classifiers for each category and aggregated the predictions (one vs. rest classification) with four models, namely Logistic Regression, Support Vector Machines (RBF), Random Forest, and XGBoost. Using a bigram term frequency-inverse document frequency (TF–IDF) encoding, the parameters for each model are tuned with a randomized search on the validation set.

Convolutional Neural Network with randomly initialized embeddings We follow the work of [4] by using Convolutional Neural Network (CNN) for multi-label text classification. In the case of text classification, CNN helps in deriving meaningful spatial signals by using filters as n-gram feature extractors. They also integrate well with word embeddings to provide information about the semantic space of the text to the model.

Regarding the intuition behind how the model works on textual data, the tokenized segments (padded to maintain the same length) are fed to the embedding layer, which outputs the embedded vectors of these tokens. We used randomly initialized word embeddings for the embedding layer, then fed them to the convolutional layer. This layer applies convolutions to the tokens by using filters that act as character/word-level n-gram detectors. This is followed by pooling operations that extract the most relevant information from the feature maps, resulting in a uni-variate vector. Therefore, the network is constrained to focus only on prominent features specific to the task at hand. Finally, the pooled outputs were fed to a series of fully connected layers with dropout to create a higher-level representation of the information, then passed to a linear layer with the same number of nodes as labels. This results in an output in target dimensions which is then passed through a sigmoid activation function to compute per-category probability scores. We used the Adam optimization algorithm to evaluate the classifier with early stopping using binary cross-entropy loss with decaying learning rate.

4.6.4 Experiments

For traditional models we tuned the models using randomized search on the validation set whereas in the case of CNN, we used an organized approach to perform an ablation study to determine what decisions, parameters or any combination of these influence the model's performance significantly. To that end, we leveraged the Optuna library [1] to conduct a series of experiments to tune the models on hyperparameters related to many aspects of the modeling process. For instance, we tuned some parameters related to the dataset pre-processing (like the token level, lower text), embedding parameters (such as the embedding type, dimensions) and others related to training and the architecture of the model (such as learning rate, number of epochs, filter size, drop out etc.). To achieve faster run times of these experiments, we utilized Nvidia CUDA library [8] support for PyTorch [10] by spinning up an Nvidia Tesla T4 instance on the Google Cloud Platform. In addition, we extensively utilized the MLflow library [2] to track all the parameters of the conducted experiments.

5 Results and Discussion

The results are shown for both union and majority-based datasets. As shown in the Figure 15 (section B.9), logistic regression gave the best results with a 0.7 f1-score while producing a balanced precision and recall. The model performed well on examples with just one label but generally tended to miss some labels in a multi-label scenario. We further performed interpretation on the Logistic Regression model results using the 'Eli5' library. Figure 12 (section B.6) shows the top 10 features with their weights for each of our target classes. Intercept (bias) of the Logistic regression model is denoted as <BIAS> in the table. Looking at the top 10 features of each class, we see that the model learned the vocabulary that is in line with the respective category's definition. For example, looking at the category 'Third Party Sharing/Collection,' the top features such as 'share', 'advertisers', 'partners', 'companies' are very relevant to this category. In addition, we also looked at individual examples to check prediction results from our model, and Figures 13 (section B.7) and 14 (section B.8) show examples of that. They show the probability of the selected example being present in different categories and highlight and score words based on their importance. Again, if we look at the category' Third Party Sharing/Collection' in the first example, we see all the words marked yellow contributed to the positive score. Words marked in the <BIAS> category contributed to the negative score.

Support Vector Machines using RBF kernel performed slightly worse than the Logistic Regression Model with a precision score of 0.74 and a recall score of 0.64. This model did not perform well on a couple of categories: 'Data Retention' and 'Practice Not Covered.' The F1 score for these categories turned out to be around 0.4, while all the other ten categories have an F1 score of more than 0.6 as shown in Figure 17 (section B.11).

We then used tree-based models and found that even after extensive hyperparameter tuning, the Random Forest did not give out good results with the recall of only 0.49 and an F1 score of 0.6 as given in Figure 16 (section B.10). One reason could be that due to the sparse nature of the TF-IDF encoded data, Random Forest fails to capture patterns as they tend to perform poorly on sparse datasets. XGBoost, on the other hand, gave an F1 score of 0.68, as shown in Figure 18 (section B.12)) which is pretty close to the best score of 0.71. With good enough hyperparameters, this model may give better results than the Logistic Regression model on these evaluation metrics.

In the case of the convolutional neural networks, we initially conducted experiments with character-level (up to 10 filter sizes) and word-level tokens (up to 5 filter sizes) using randomly initialized embeddings. On the surface, they worked comparably on the weighted and micro averages on both datasets, as shown in the Figure 19 (section B.13). However, the character-level model tends to produce balanced values of precision and recall. In contrast, the word-level model prioritizes precision and fails to reduce a lot of false negatives, thus leading to a lower recall. Also, we notice a relative improvement in the average F1 scores in the majority dataset compared to the union-based. Additional analysis of the results are detailed in the appendix A.2.

Overall, despite their strengths and weaknesses, we believe all the models performed well to produce satisfactory results. However, the model fails to effectively capture essential distinctions between the categories due to the limitations in how the data is represented (TF-IDF and random embeddings). Thus, we believe there is a need to scale up the data and model complexities further to gain improved performances on the privacy policy text classification.

6 Future Directions

We developed baseline models with extensive pre-processing and tuning and obtained results with reliable predictions. Leveraging the learnings from this phase, we plan to supplement our efforts in scaling up the data and model complexities further in the form of word embeddings and complex architectures. To that end, we seek to employ pre-trained word embeddings and train custom embeddings on a corpus of 1M policies [13] to provide more domain-specific context for the models to learn from. We would also like to extend this further by employing recurrent neural network architectures (such as the Long Short Term Memory (LSTM), Gated Recurrent Units (GRU)), as well as transformer models like BERT (Bidirectional Encoder Representations from Transformers) [3] in an effort to achieve competitive scores. We expect the above methods to take up a quarter of the next phase before we move on to implement the QUERY module.

7 Statement of Contributions

Shubhanshu Gupta and Kaushik Nishtala contributed equally to the project. Shubhanshu Gupta conceived and planned the experiments to perform data pre-processing. In addition, he also carried out experiments on traditional models and worked on building the back-end for the web framework. Kaushik Nishtala contributed to dataset preparation, the interpretation of the results, designed the model and the computational framework for CNN, and contributed to the front-end development of the web framework.

Github The code used for analysis, experiments and the web framework is provided in a repository at `https://github.com/Kau5h1K/ds5500-userprivacy`.

References

[1] T. AKIBA, S. SANO, T. YANASE, T. OHTA, AND M. KOYAMA, *Optuna: A next-generation hyperparameter optimization framework*, 2019.

[2] A. CHEN, A. CHOW, A. DAVIDSON, A. DCUNHA, A. GHODSI, S. A. HONG, A. KONWINSKI, C. MEWALD, S. MURCHING, T. NYKODYM, P. OGILVIE, M. PARKHE, A. SINGH, F. XIE, M. ZAHARIA, R. ZANG, J. ZHENG, AND C. ZUMAR, *Developments in mlflow: A system to accelerate the machine learning lifecycle*, in Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning, DEEM'20, New York, NY, USA, 2020, Association for Computing Machinery.

[3] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019.

[4] H. HARKOUS, K. FAWAZ, R. LEBRET, F. SCHAUB, K. G. SHIN, AND K. ABERER, *Polisis: Automated analysis and presentation of privacy policies using deep learning*, 2018.

[5] F. LIU, S. WILSON, P. STORY, S. ZIMMECK, AND N. M. SADEH, *Towards automatic classification of privacy policy text*, 2017.

[6] G. MOHANDAS, *Embeddings - made with ml*, (2021).

[7] N. MOUSAVI NEJAD, P. JABAT, R. NEDELCHEV, S. SCERRI, AND D. GRAUX, *Establishing a strong baseline for privacy policy classification*, in ICT Systems Security and Privacy Protection, M. Hölbl, K. Rannenberg, and T. Welzer, eds., Cham, 2020, Springer International Publishing, pp. 370–383.

[8] J. NICKOLLS, I. BUCK, M. GARLAND, AND K. SKADRON, *Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for?*, Queue, 6 (2008), p. 40–53.

[9] J. A. OBAR AND A. OELDORF-HIRSCH, *The biggest lie on the internet: ignoring the privacy policies and terms of service policies of social networking services*, Information, Communication & Society, 23 (2020), pp. 128–147.

[10] A. PASZKE, S. GROSS, F. MASSA, A. LERER, J. BRADBURY, G. CHANAN, T. KILLEEN, Z. LIN, N. GIMELSHEIN, L. ANTIGA, A. DESMAISON, A. KOPF, E. YANG, Z. DEVITO, M. RAISON, A. TEJANI, S. CHILAMKURTHY, B. STEINER, L. FANG, J. BAI, AND S. CHINTALA, *Pytorch: An imperative style, high-performance deep learning library*, in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[11] F. SEBASTIANI, *Machine learning in automated text categorization*, ACM Computing Surveys, 34 (2002), p. 1–47.

[12] K. SECHIDIS, G. TSOUMAKAS, AND I. VLAHAVAS, *On the stratification of multi-label data*, in Machine Learning and Knowledge Discovery in Databases, D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 145–158.

[13] M. SRINATH, S. WILSON, AND C. L. GILES, *Privacy at scale: Introducing the privaseer corpus of web privacy policies*, 2020.

[14] P. SZYMAŃSKI AND T. KAJDANOWICZ, *A scikit-based Python environment for performing multi-label classification*, ArXiv e-prints, (2017).

[15] S. WILSON, F. SCHAUB, A. A. DARA, F. LIU, S. CHERIVIRALA, P. GIOVANNI LEON, M. SCHAARUP ANDERSEN, S. ZIMMECK, K. M. SATHYENDRA, N. C. RUSSELL, T. B. NORTON, E. HOVY, J. REIDENBERG, AND N. SADEH, *The creation and analysis of a website privacy policy corpus*, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, Aug. 2016, Association for Computational Linguistics, pp. 1330–1340.

A Appendix

A.1 Visualization framework: sample set of questions

The produced plots may answer holistic questions such as

- How is my health/location information used?
- What information collection can I opt-out from?
- Why is my information shared?
- What information is shared with external third parties?
- What choices are available/not available?
- To what uses of information do opt-outs apply?
- Can I view/edit/delete/export my account?
- How long is my information retained for advertising or marketing purposes?
- How is my information protected?

A.2 Supplementary results

Regarding label-specific performances of the models, we noticed that the model failed to perform well in classifying the "Practice Not Covered" category. This could be due to the diversity of vocabulary found in the corresponding paragraphs. This ambiguity is further fueled in the case of the majority-based dataset when the prediction scores are zeros, as shown in the Figure 19 (section B.13). Furthermore, almost all models perform equally well on the Do Not Track class despite having very few samples in the data. This could be attributed to a specific set of vocabulary found in paragraphs related to this category, including words like track and signal. Also, looking at the Fleiss' Kappa values mentioned earlier, the best human annotation agreement was achieved on this category with the statistic equal to 91%, which indicates that our models reasonably mimic and reproduce human thought processes.

Additionally, Figure 4 shows the F1 scores of all categories plotted against the number of samples in the union-dataset for these categories. Generally, as the number of samples increases, we expect the model to produce more stable results. We notice that the categories with more than 200 samples show relatively stable F1 scores except the Privacy Contact Information category. The model fails to learn the terminology specific to this category despite its high sample occurrence.

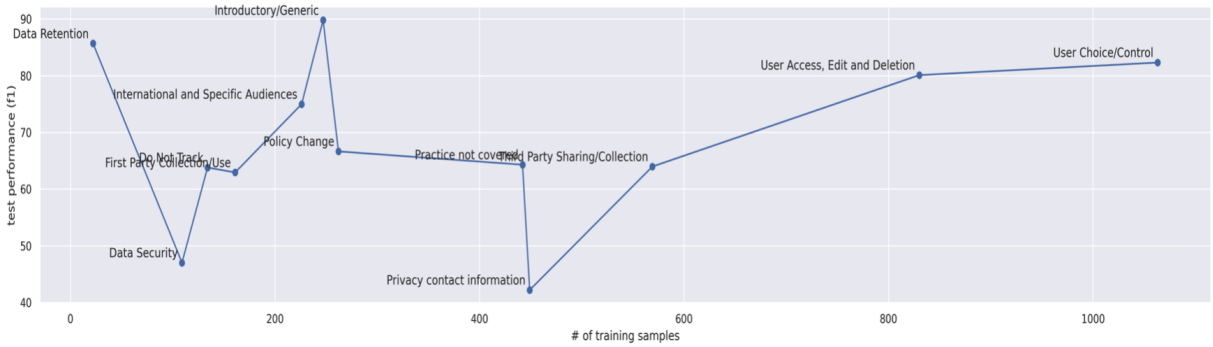

Figure 4: F1 scores of each category against the number of training samples.

Finally, we show sample predictions of the CNN model (word-level) on the union-dataset. We believe that looking at false positives and false negatives would help us find any potential mislabeling cases in the

annotations and understand what aspects of the dataset the model has failed to learn. From the Figure 5 that shows the false positive prediction, we see that the model classifies the segment related to Third Party Sharing/Collection as both First Party Collection/Use and Third Party Sharing/Collection. We imagine this has something to do with the model failing to effectively capture the distinction between these categories since they have a similar set of terminology.

```
=== False positives ===
  aggregate information non personally identifiable we share aggregated demographic information
with our partners and advertisers this is not linked to any personally identifiable information
    true: ['Third Party Sharing/Collection']
    pred: ['First Party Collection/Use', 'Third Party Sharing/Collection']
```

Figure 5: A sample false positive prediction of CNN (word-level) on union-dataset.

Lastly, the false negative prediction shows a potential mislabeling with the First Party Collection/Use. However, the segment is specifically about User Choice/Control, which the model predicted correctly.

```
=== False negatives ===

  newsletter if a user wishes to subscribe to our newsletter we ask for contact information suc
h as name and email address out of respect for our users privacy we provide a way to opt out of
these communications please see the choice and opt out sections
    true: ['First Party Collection/Use']
    pred: ['User Choice/Control']
```

Figure 6: A sample false negative prediction of CNN (word-level) on union-dataset.

B Figures

B.1 Annotation scheme

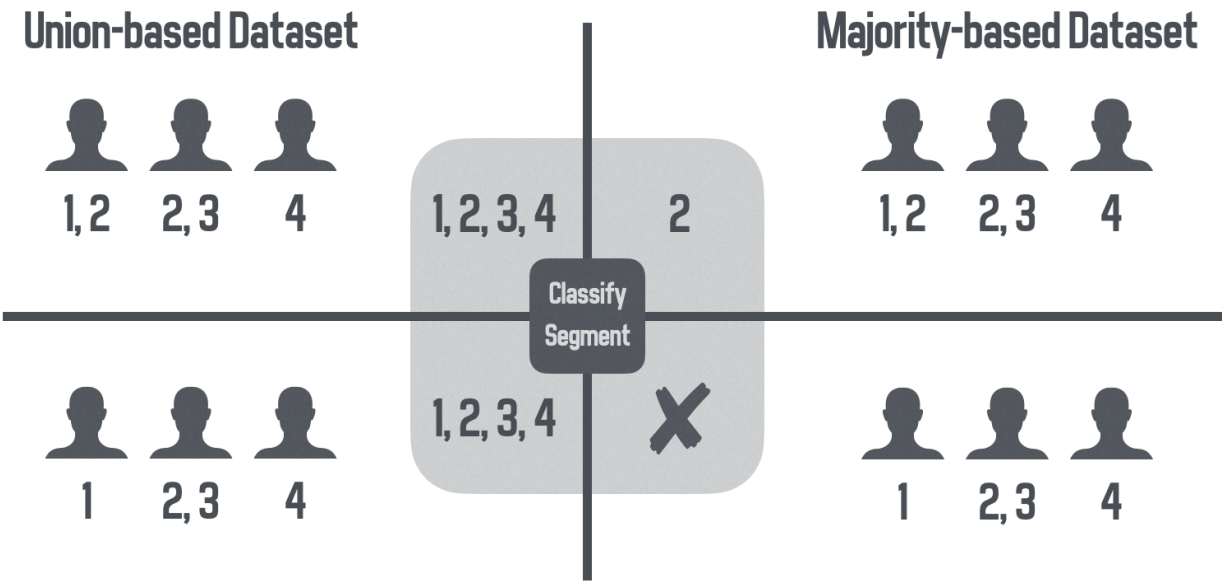

Figure 7: Figure shows the annotation ruling of the compiled gold standard datasets.

B.2 Data Preprocessing

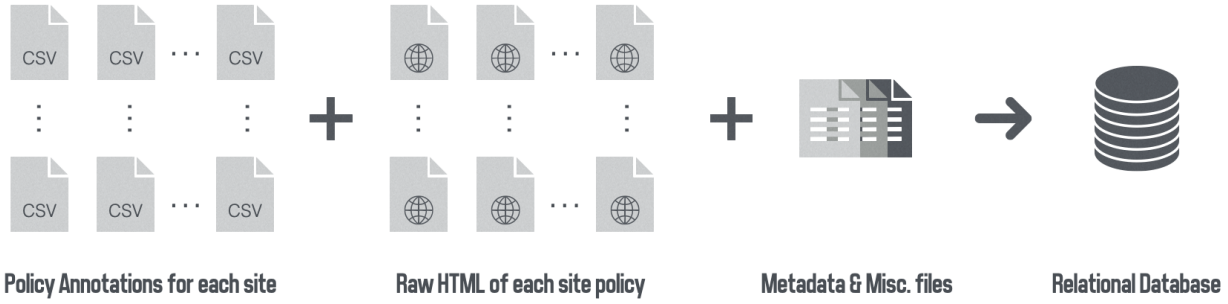

Figure 8: Various pre-processing tasks were performed to build a relational database for the visualization framework.

B.3 Entity Relationship Diagram

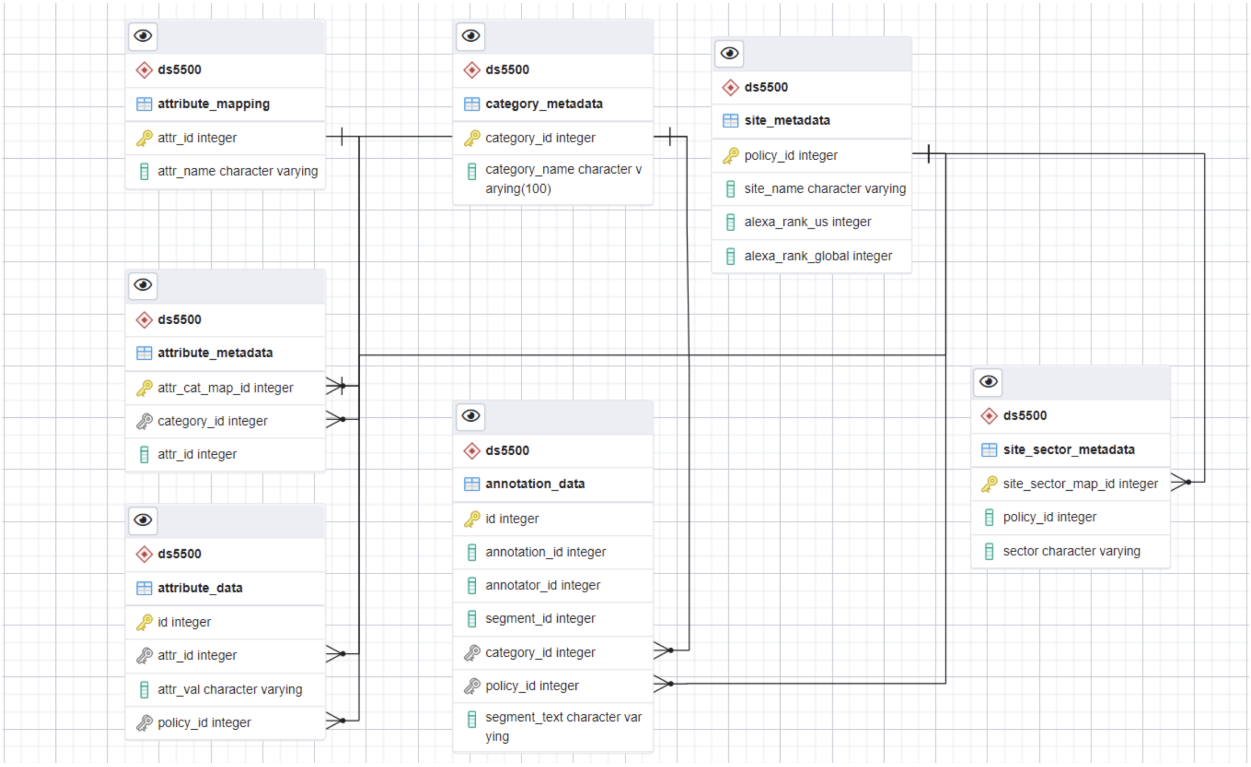

Figure 9: The schematic representation of the processed dataset that serves the visualization framework.

B.4 Naive stratification

	(9,)	(4,)	(5,)	(11,)	(6,)	(10,)	(8,)	(1,)	(7,)	(3,)	(0,)	(2,)
Train	850	233	576	450	138	157	216	257	460	1062	106	23
Val	840	322	592	373	135	163	228	266	410	1017	98	18
Test	728	238	513	476	116	177	270	284	438	1115	135	23

Figure 10: The adjusted label proportions of naive splits across train, test and validation. The columns denote the encoded data practice categories.

B.5 Iterative stratification

	(0,)	(3,)	(5,)	(11,)	(8,)	(7,)	(9,)	(6,)	(1,)	(10,)	(2,)	(4,)
Train	109	1063	569	442	226	449	830	134	262	161	22	247
Val	116	1064	569	443	228	448	830	135	266	163	32	247
Test	102	1064	569	443	224	452	830	135	261	158	14	247

Figure 11: The adjusted label proportions of iterative splits across train, test and validation. The columns denote the encoded data practice categories.

B.6 Logistic regression interpretation: Example 1

y=First Party Collection/Use top features		**y=Third Party Sharing/Collection** top features		**y=Privacy contact information** top features		**y=Policy Change** top features		**y=Do Not Track** top features		**y=User Choice/Control** top features	
Weight?	**Feature**	**Weight?**	**Feature**	**Weight?**	**Feature**	**Weight?**	**Feature**	**Weight?**	**Feature**	**Weight?**	**Feature**
+16.744	collect	+27.070	share	+21.994	questions	+36.418	changes	+134.022	signals	+24.892	opt
+16.645	may use	+22.363	disclose	+19.908	please	+20.013	revised	+121.900	dnt	+22.342	unsubscribe
+12.229	uses	+20.309	companies	+19.740	contact	+19.492	updated october	+84.742	track	+16.065	option
+12.070	use	+17.517	advertisers	+15.551	com			+62.955	track signals	+15.043	consent
+11.795	participate	+16.633	shared	+13.437	eu	+17.474	last			+12.472	want
+11.512	emails	+16.535	sell	+11.662	contacting	+14.102	changes privacy	+52.392	response signals	+12.298	disable
+11.253	commercial	+16.471	partners	+10.923	suite					+11.733	choices
… 5476 more positive …		+14.795	partner	+10.737	please send	+13.742	notice	+44.963	meaning	+11.608	written
… 4531 more negative …		+12.951	providers may	+10.601	contacting us	+12.870	modified june	+44.385	track dnt	+11.462	options
-11.174	please	*… 5053 more positive …*		+10.454	choose opt	+12.295	last updated	+43.983	currently	*… 5117 more positive …*	
-11.417	secure	*… 3042 more negative …*		*… 3141 more positive …*		+12.164	policy time	+35.720	assign	*… 2334 more negative …*	
-14.348	privacy	-12.960	security	*… 1153 more negative …*		+11.821	prior		meaning	-11.341	track
						… 1642 more positive …		+35.720	potential dnt		
						… 415 more negative …		*… 154 more positive …*			
								… 703 more negative …			

Figure 12: Logistic regression - Top ranked vocabulary for six selected categories.

B.7 Logistic regression interpretation: Example 2

y=First Party Collection/Use (probability **0.921**, score **2.463**) top features

Contribution?	**Feature**
+4.876	Highlighted in text (sum)
-2.413	<BIAS>

collection personally identifiable information social media sites addition , interact atlantic property page account social media platform , facebook , twitter , tumblr , linkedin , may collect personally identifiable information make available us page account including social media account id . however , comply privacy policies corresponding social media platform collect store personally identifiable information permitted collect social media platforms . choose link login atlantic account social networking service , atlantic service may share certain information activities . consent , also may share information activities , including view sites , social networks users .

y=Third Party Sharing/Collection (probability **0.995**, score **5.364**) top features

Contribution?	**Feature**
+8.626	Highlighted in text (sum)
-3.262	<BIAS>

collection personally identifiable information social media sites addition , interact atlantic property page account social media platform , facebook , twitter , tumblr , linkedin , may collect personally identifiable information make available us page account including social media account id . however , comply privacy policies corresponding social media platform collect store personally identifiable information permitted collect social media platforms . choose link login atlantic account social networking service , atlantic service may share certain information activities . consent , also may share information activities , including view sites , social networks users .

Figure 13: Logistic regression label annotations for First Party Collection/Use and Third Party Sharing/Collection categories.

B.8 Logistic regression interpretation: Example 3

y=Privacy contact information (probability **0.001**, score **-7.457**) top features

Contribution[?]	Feature
-2.549	Highlighted in text (sum)
-4.908	<BIAS>

collection personally identifiable information social media sites addition , interact atlantic property page account social media platform , facebook , twitter , tumblr , linkedin , may collect personally identifiable information make available us page account including social media account id . however , comply privacy policies corresponding social media platform collect store personally identifiable information permitted collect social media platforms . choose link login atlantic account social networking service , atlantic service may share certain information activities . consent , also may share information activities , including view sites , social networks users .

y=Policy Change (probability **0.000**, score **-7.690**) top features

Contribution[?]	Feature
-1.229	Highlighted in text (sum)
-6.461	<BIAS>

collection personally identifiable information social media sites addition , interact atlantic property page account social media platform , facebook , twitter , tumblr , linkedin , may collect personally identifiable information make available us page account including social media account id . however , comply privacy policies corresponding social media platform collect store personally identifiable information permitted collect social media platforms . choose link login atlantic account social networking service , atlantic service may share certain information activities . consent , also may share information activities , including view sites , social networks users .

Figure 14: Logistic regression label annotations for Privacy contact information and Policy change categories.

B.9 Logistic regression model performance results

	precision	recall	f1-score	support	precision	recall	f1-score	support
					UNION			
Data Retention	0.52	0.57	0.54	23	0.31	0.45	0.37	11
Data Security	0.71	0.79	0.75	56	0.79	0.71	0.75	31
Do Not Track	0.71	0.83	0.77	6	0.44	1.00	0.62	4
First Party Collection/Use	0.80	0.79	0.79	228	0.81	0.80	0.80	181
International and Specific Audiences	0.89	0.77	0.83	53	0.89	0.76	0.82	45
Introductory/Generic	0.64	0.57	0.60	122	0.68	0.67	0.68	58
Policy Change	0.70	0.72	0.71	29	0.65	0.83	0.73	18
Practice not covered	0.49	0.44	0.47	97	0.25	0.05	0.08	20
Privacy contact information	0.72	0.73	0.73	49	0.64	0.81	0.71	31
Third Party Sharing/Collection	0.78	0.80	0.79	178	0.79	0.88	0.83	142
User Access, Edit and Deletion	0.76	0.65	0.70	34	0.77	0.74	0.76	23
User Choice/Control	0.72	0.66	0.69	95	0.71	0.77	0.74	53
micro avg	0.72	0.70	0.71	970	0.75	0.77	0.76	617
macro avg	0.70	0.69	0.70	970	0.65	0.71	0.66	617
weighted avg	0.72	0.70	0.71	970	0.74	0.77	0.75	617
samples avg	0.75	0.77	0.73	970	0.75	0.79	0.75	617

MAJORITY

Figure 15: Precision, Recall and F1 scores for the Logistic regression model on the two gold standard datasets (in decimals) on validation

B.10 Random Forest model performance results

	precision	recall	f1-score	support	precision	recall	f1-score	support
Data Retention	0.67	0.09	0.15	23	0.00	0.00	0.00	11
Data Security	0.89	0.45	0.60	56	0.94	0.48	0.64	31
Do Not Track	0.00	0.00	0.00	6	0.00	0.00	0.00	4
First Party Collection/Use	0.88	0.68	0.77	228	0.90	0.71	0.79	181
International and Specific Audiences	0.94	0.62	0.75	53	0.96	0.51	0.67	45
Introductory/Generic	0.81	0.41	0.54	122	0.94	0.26	0.41	58
Policy Change	0.76	0.45	0.57	29	1.00	0.39	0.56	18
Practice not covered	0.80	0.12	0.21	97	0.00	0.00	0.00	20
Privacy contact information	1.00	0.47	0.64	49	0.81	0.42	0.55	31
Third Party Sharing/Collection	0.89	0.71	0.79	178	0.90	0.68	0.78	142
User Access, Edit and Deletion	1.00	0.21	0.34	34	0.00	0.00	0.00	23
User Choice/Control	0.76	0.29	0.42	95	0.85	0.43	0.58	53
micro avg	0.87	0.49	0.63	970	0.90	0.52	0.66	617
macro avg	0.78	0.38	0.48	970	0.61	0.32	0.41	617
weighted avg	0.85	0.49	0.60	970	0.82	0.52	0.62	617
samples avg	0.69	0.58	0.61	970	0.57	0.55	0.55	617

UNION MAJORITY

Figure 16: Precision, Recall and F1 scores for the Random Forest model on the two gold standard datasets (in decimals) on validation

B.11 Support Vector Machines model performance results

	precision	recall	f1-score	support	precision	recall	f1-score	support
Data Retention	0.56	0.39	0.46	23	0.60	0.27	0.37	11
Data Security	0.78	0.71	0.75	56	0.81	0.68	0.74	31
Do Not Track	1.00	0.67	0.80	6	1.00	0.50	0.67	4
First Party Collection/Use	0.84	0.77	0.80	228	0.82	0.80	0.81	181
International and Specific Audiences	0.89	0.75	0.82	53	0.92	0.73	0.81	45
Introductory/Generic	0.63	0.55	0.59	122	0.77	0.64	0.70	58
Policy Change	0.73	0.76	0.75	29	0.84	0.89	0.86	18
Practice not covered	0.49	0.33	0.40	97	0.40	0.10	0.16	20
Privacy contact information	0.74	0.69	0.72	49	0.74	0.74	0.74	31
Third Party Sharing/Collection	0.81	0.72	0.76	178	0.80	0.84	0.82	142
User Access, Edit and Deletion	0.81	0.50	0.62	34	1.00	0.70	0.82	23
User Choice/Control	0.70	0.55	0.62	95	0.78	0.74	0.76	53
micro avg	0.75	0.64	0.69	970	0.81	0.74	0.77	617
macro avg	0.75	0.62	0.67	970	0.79	0.63	0.69	617
weighted avg	0.74	0.64	0.69	970	0.80	0.74	0.76	617
samples avg	0.75	0.72	0.70	970	0.77	0.77	0.75	617

UNION MAJORITY

Figure 17: Precision, Recall and F1 scores for the Support Vector Machines model on the two gold standard datasets (in decimals) on validation

B.12 XGBoost model performance results

	precision	recall	f1-score	support	precision	recall	f1-score	support
Data Retention	0.59	0.43	0.50	23	0.50	0.18	0.27	11
Data Security	0.80	0.62	0.70	56	0.74	0.65	0.69	31
Do Not Track	0.80	0.67	0.73	6	0.50	1.00	0.67	4
First Party Collection/Use	0.82	0.75	0.78	228	0.83	0.77	0.80	181
International and Specific Audiences	0.97	0.72	0.83	53	0.96	0.60	0.74	45
Introductory/Generic	0.70	0.50	0.58	122	0.87	0.57	0.69	58
Policy Change	0.74	0.69	0.71	29	0.78	0.78	0.78	18
Practice not covered	0.57	0.29	0.38	97	0.25	0.05	0.08	20
Privacy contact information	0.85	0.71	0.78	49	0.69	0.65	0.67	31
Third Party Sharing/Collection	0.80	0.75	0.77	178	0.81	0.76	0.78	142
User Access, Edit and Deletion	0.83	0.44	0.58	34	0.86	0.52	0.65	23
User Choice/Control	0.80	0.52	0.63	95	0.81	0.66	0.73	53
micro avg	0.79	0.62	0.69	970	0.81	0.67	0.73	617
macro avg	0.77	0.59	0.66	970	0.72	0.60	0.63	617
weighted avg	0.78	0.62	0.68	970	0.80	0.67	0.72	617
samples avg	0.75	0.70	0.69	970	0.70	0.69	0.69	617

UNION MAJORITY

Figure 18: Precision, Recall and F1 scores for the XGBoost model on the two gold standard datasets (in decimals) on validation

B.13 Convolutional Neural Network model performance results

Character level Word level

Union Majority Union Majority

	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
Data Retention	0.800000	0.347826	0.484848	23.0	1.000000	0.083333	0.153846	12.0	1.000000	0.173913	0.296296	23.0	1.000000	0.083333	0.153846	12.0
Data Security	0.730769	0.666667	0.697248	57.0	0.923077	0.774194	0.842105	31.0	0.853659	0.614035	0.714286	57.0	0.956522	0.709677	0.814815	31.0
Do Not Track	1.000000	1.000000	1.000000	5.0	1.000000	0.800000	0.888889	5.0	1.000000	0.200000	0.333333	5.0	1.000000	0.800000	0.888889	5.0
First Party Collection/Use	0.710714	0.872807	0.783465	228.0	0.783505	0.839779	0.810667	181.0	0.835681	0.780702	0.807256	228.0	0.804233	0.839779	0.821622	181.0
International and Specific Audiences	0.960000	0.905660	0.932039	53.0	0.900000	0.800000	0.847059	45.0	1.000000	0.849057	0.918367	53.0	0.970588	0.733333	0.835443	45.0
Introductory/Generic	0.636364	0.573770	0.603448	122.0	0.767442	0.559322	0.647059	59.0	0.764706	0.532787	0.628019	122.0	0.833333	0.508475	0.631579	59.0
Policy Change	0.850000	0.586207	0.693878	29.0	0.937500	0.833333	0.882353	18.0	0.933333	0.482759	0.636364	29.0	0.833333	0.833333	0.833333	18.0
Practice not covered	0.515152	0.350515	0.417178	97.0	0.000000	0.000000	0.000000	20.0	0.606061	0.206186	0.307692	97.0	1.000000	0.050000	0.095238	20.0
Privacy contact information	0.833333	0.625000	0.714286	48.0	0.791667	0.633333	0.703704	30.0	0.911765	0.645833	0.756098	48.0	0.800000	0.533333	0.640000	30.0
Third Party Sharing/Collection	0.683486	0.837079	0.752525	178.0	0.698718	0.767606	0.731544	142.0	0.829114	0.735955	0.779762	178.0	0.787879	0.732394	0.759124	142.0
User Access, Edit and Deletion	0.857143	0.529412	0.654545	34.0	0.882353	0.652174	0.750000	23.0	0.937500	0.441176	0.600000	34.0	0.875000	0.304348	0.451613	23.0
User Choice/Control	0.686275	0.736842	0.710660	95.0	0.809524	0.629630	0.708333	54.0	0.698795	0.610526	0.651685	95.0	0.861111	0.574074	0.688889	54.0
micro avg	0.707216	0.707946	0.707581	969.0	0.783688	0.712903	0.746622	620.0	0.820055	0.616099	0.703595	969.0	0.828685	0.670968	0.741533	620.0
macro avg	0.771936	0.669315	0.703677	969.0	0.791149	0.614392	0.663797	620.0	0.864218	0.522744	0.619097	969.0	0.893500	0.558507	0.634533	620.0
weighted avg	0.708186	0.707946	0.697765	969.0	0.769452	0.712903	0.728699	620.0	0.814190	0.616099	0.685493	969.0	0.842847	0.670968	0.720337	620.0
samples avg	0.743398	0.770745	0.720586	969.0	0.721063	0.737192	0.712081	620.0	0.799002	0.701585	0.713037	969.0	0.719165	0.695762	0.693169	620.0

Figure 19: Precision, Recall and F1 scores for the Convolutional Neural Network model on the two gold standard datasets (in decimals) with tuned epochs on validation